Slow Growing Subdivision (SGS) in Any Dimension: Towards Removing the Curse of Dimensionality

V. Pascucci [†]

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory, Livermore, CA

Abstract

In recent years subdivision methods have been one of the most successful techniques applied to the multi-resolution representation and visualization of surface meshes. Extension these techniques to the volumetric case would enable their use in a broad class of applications including solid modeling, scientific visualization and mesh generation. Unfortunately, major challenges remain unsolved both in the generalization of the combinatorial structure of the refinement procedure and in the analysis of the smoothness of the limit mesh.

In this paper we mainly tackle the first part of the problem introducing a subdivision scheme that generalizes to 3D and higher dimensional meshes without the excessive vertex proliferation typical of tensor-product refinements. The main four qualities of our subdivision procedure are: (i) the rate of refinement does not grow with the dimension of the mesh, (ii) adaptive refinement of the mesh is possible without introducing special temporary cell decompositions, (iii) the cells of the base meshes can have virtually unrestricted topology, and (iv) "sharp" features of different dimensions can be incorporated naturally.

We use a narrow averaging mask that is applied to the vertices of the mesh and/or to eventual functions defined on the mesh. The general study of the limit smoothness of the approach requires new analysis techniques that are beyond the scope of this paper.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations. *Volumetric meshes, recursive subdivision methods.*

1. INTRODUCTION

Modern scanning devices, modeling systems, and computer simulations gives rise to surface and volume meshes of ever increasing resolution. Real-time display and transmission of this sheer amount of data is a challenging task requiring to generate approximations of minimal size with respect to given error bounds. Great progress has been made in the case of surface meshes, with the use of multi-resolution representations based on subdivision surfaces. Unfortunately the generalization of such methods to the volumetric case is not straightforward in general, and remains almost entirely limited to the case of tensor-product generalization of 1D subdivisions. The excessive rate of refinement of tensor product

schemes and the lack of good extensions to general unstructured meshes limits severely the use of such techniques in fundamental application areas like solid modeling, scientific visualization and mesh generation.

One trend in multi-resolution surface generation is the design of methods based on wavelet functions [3, 14, 22, 24]. The wavelet analysis has the advantage of generating directly a multi-resolution data-structure with guaranteed error bounds. The basic ingredient needed for wavelet analysis is the construction of nested function spaces which are best associated with the connectivity of subdivision surfaces. This restricts the class of meshes that can be processed, requiring eventual re-meshing of the input. Hybrid approaches can be designed to take advantage of the quality of wavelet analysis while using a more general simplification scheme [11].

The general framework of wavelet analysis is formalized independently of the intrinsic/embedding dimension of the

---

geometric object. This enables multi-resolution representation and analysis for volumetric data [17, 21, 23]. The definition of subdivision connectivity in the case of general case of unstructured meshes will allow a more extensive use of wavelet representation schemes.

Solutions similar to subdivision methods have been designed in the meshing community for adaptive refinement of triangular meshes [2]. Rivara's edge bisection approach is one of the simplest and most flexible of these [20]. A unique subdivision template is used to subdivide the cells of a 2D mesh until a given resolution is achieved. The approach generalizes to 3D tetrahedralizations [18, 20] and to higher dimensions by performing the refinement process successively from the lower dimensional simplices (edges) to the higher dimensional ones. This scheme is is combinatorially equivalent to our Slow Growing Subdivision (*SGS* from now on) for a particular tetrahedralization of rectilinear grids.

Recursive subdivision schemes automatically produce hierarchical multi-resolution representations. This enables, for example, multi-resolution editing techniques [29]. The quality of the mesh generated by subdivision schemes depends on the averaging masks used. For triangulated domains Loop [13] provides an approximating subdivision scheme converging to a surface that is C^2 almost everywhere, with exception of the extraordinary vertices (degree not equal to six) where the surface is C^1. The butterfly subdivision scheme [7] converges to an interpolating surface that is C^1 everywhere, except for extraordinary points with degree three or higher than seven. A modified version has been proposed [30] that converges to a C^1 surface everywhere. For subdivision of piecewise quadrilateral domains, one can use the Catmull-Clark scheme [5] or the interpolating scheme by Kobbelt [9] to build smooth approximations of a coarse mesh. Biermann et al. [4] have improved the normal control mechanism [8] for Catmull-Clark and Loop subdivisions.

Kobbelt [10] has been the first to address the problem of excessive proliferation is a refinement procedure by introducing the $\sqrt{3}$-subdivision scheme that increases the number of vertices in the mesh at slower rate than previous approaches. This approach has been improved by Vehlo et al. [28, 27, 25, 26] and independently by Ducheneau et al. [6] with an edge bisection refinement that can be categorized as $\sqrt{2}$ subdivision scheme (the 2D case of our *SGS*). This problem gets worse as the dimension d of the mesh increases. The tensor product generalization of the surface subdivision techniques [1, 15, 19] leads to prohibitive refinement rate since a uniform smoothing step requires increasing the number of vertices by a factor that grows exponentially with the dimension d of the mesh. This rapid increase of the model complexity can quickly make the use of such schemes impractical. Adaptive refinement also requires special rules to temporarily partition any cell that connects regions at different levels of resolution.

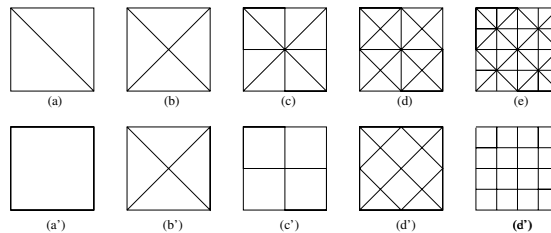The *SGS* introduced here roughly doubles the number of

Figure 1: *4-8 recursive subdivision. (a-e) Classical longest edge bisection of a rectilinear grid. (a'-e') Equivalent $\sqrt{2}$ subdivision where pairs of adjacent triangles are merged into one square.*

vertices independently of the intrinsic dimension of the input mesh. In 3D, for example, each *SGS* refinement doubles the number of vertices instead of increasing them by a factor of eight (as in a tensor product scheme). Linsen et al. show in [12] how to define a wavelet hierarchy on the basis of this subdivision scheme. Four major characteristics make *SGS* attractive for a practical purposes:

Slow The rate of refinement (new vertices introduced in the mesh) is nearly independent of the dimensionality of the mesh.

General. The scheme applies for any complex of polyhedral cells such that each cell is topologically a ball. This includes tetrahedral meshes, curvilinear hexahedral meshes and any mesh with convex elements.

Adaptive. The scheme naturally includes a mechanism for locally adapted refinements and for handling lower dimensional "sharp" features. There is no need to introduce separate classes of cells to connect regions at different level of resolution.

Small support. The masks that we use for the basic smoothing scheme are as small or smaller than those of the Catmull-Clark subdivision.

2. REVIEW OF THE SURFACE SUBDIVISION SCHEME

The $\sqrt{2}$ recursive subdivision scheme, also known as 4-8 subdivision surface [28], follows the edge bisection refinement rules introduced by Rivara in [20]. Figure 1 (a-e) shows the subdivision scheme for a rectilinear grid. The base mesh is a square divided into two triangles. At any refinement each triangle bisected at the middle of its longest edge. The 4-8 subdivision rule follows these combinatorial rules and adds an averaging step that repositions the vertices on the surface. To maintain the same combinatorial subdivision structure one has to use the "oldest bisection" rule. In each triangle one has to bisect the edge that was unaltered in the previous refinement (there is only one).

Figure 1(a'-e') shows the equivalent subdivision strategy for quadrilateral elements [6]. Each refinement is performed
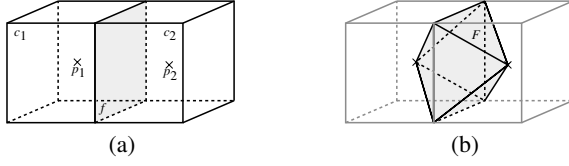
Figure 2: *3D cell refinement from tier 0 to tier 1. (a) The two cells c_1 and c_2 in tier 0. Their centers p_1 and p_2 are marked with two crosses. Their adjacency facet f is highlighted in gray. (b) The cell F of tier 1 (in gray) is the union of the pyramids $p_1 \triangleleft f$ and $p_2 \triangleleft f$.*

by inserting a point at the center of each *diamond* and splitting the diamond into four triangles. Then each pair of triangles adjacent along an old edge are merged into a new diamond.

The advantage of this scheme is that doubling the resolution of a rectilinear grid is performed in two steps instead of one. In the following sections we show how the *SGS* generalizes this procedures to meshes of any dimension and with cells of virtually any type. The resolution of a d-grid is doubled in d subdivision steps with an averaging mask applied at each step.

3. SLOW GROWING SUBDIVISION

We organize the subdivision process of the *SGS* scheme into levels and tiers. Each level l has four tiers, from 0 to 3, where tier 3 of level l is coincident with tier 0 of level $l+1$. This naming convention is used to maintain the comparison with classical tensor product subdivisions that would refine directly a mesh from tier 0 one level l to tier 0 of level $l+1$. In the *SGS* scheme each refinement is a transition form tier i to tier $i+1$. At tier 3 the level is increased by one and the tier is reset to 0. We denote cells, facets, edges and vertices of the with the symbols c_i, f_i, e_i and v_i respectively.

3.1. Basic Subdivision Rules

In the present context we consider only the combinatorial structure of the scheme. For example we qualify a vertex as the "center" of a cell/face just to give an intuitive notion of the relation between the point and the cell but we do not refer necessarily to its actual geometric location, which may change depending on the choice of the averaging coefficients.

From tier 0 to tier 1 For each cell c_i in the input mesh its center p_i is selected. The cell c_i, having n facets, is decomposed into n pyramidal cells by connecting the center p_i with all its facets. Let's denote by $p \triangleleft f$ the pyramid built by connecting p with a facet f. For each pair of cells c_i, c_j, adjacent along a facet f, a new cell F is created by merging the pyramid $p_i \triangleleft f$ with the pyramid $p_j \triangleleft f$:

$$F = (p_i \triangleleft f) \cup (p_j \triangleleft f), \qquad \text{with} \quad f = c_i \cap c_j.$$

Figure 2 shows the construction of F form c_1 and c_2.

From tier 1 to tier 2. Consider a cell F of tier 1 and its center q. Let g_i be the facets of F that do not belong to tier 0 (for non-sharp F all the facets are of tier 1). We decompose F into a set of pyramids each given by $q \triangleleft g_i$. If F is a sharp cell, its center q_k is coincident with the center of its facet f of tier 0. In this way we handle directly boundary cases and 2-dimensional sharp features.

Note that each pyramid $q \triangleleft g_i$ contains exactly one edge e_j of tier 0. After each tier-1 cell is split all the pyramids incident to the same edge e are merged into a cell E. All the cells built in this way form the mesh of tier 2. Figure 3 shows the construction of one cell of tier 2. The coarse mesh has four cells all incident to an edge e (Figure 3a). Four cells of tier 1 are built by merging pairs face pyramids (Figure 3b). Each tier-1 cell is then decomposed into four pyramids, of which we select only two incident to e Figure 3(c). The eight pyramids selected (two per cell) are finally merged into one cell E of tier 2, (Figure 3d).

From tier 2 to tier 3. As in the previous two steps one determines the center r of any cell E. Each cell E is then partitioned by joining r with each facet of E. As usual, for sharp cells the point r should be considered as the center of e and is shared among all the cells around e.

The last merging step is among cells that are incident both to a vertex v and a cell center p. Note that, during this last merging step, one must remove any spurious edge introduced during the refinement procedure. In particular any edge connecting two vertices introduced in tiers that differ by more than one are removed. Figure 4 shows one cell of tier 0 (a) and one of its descendants of tier 3 (b). Figure 4(c) shows the refined cell with the spurious edges (facet diagonals) connecting the vertex of tier 0 with vertices of tier 2 or the vertex of tier 1 with vertices of tier 3.

3.2. Adaptive Refinement, Sharp Features and Boundary Cells

We handle in a unified way sharp features, boundary cells and adaptive refinements, simply by not performing the merge stage of the subdivision procedure.

For example in the refinement form tier 0 to tier 1 the boundary cases allow to build only one half of the cell F. Similarly, for 2-dimensional sharp features (surfaces that need to be preserved during the refinement) we build two halves $F' = p_i \triangleleft f$ and $F'' = p_j \triangleleft f$ without merging them. In such cases the boundary pyramid F (or the two halves F' and F'') are called "sharp" since they maintain the facet f from tier 0. One can also build only one half of F to achieve adaptive refinement. In fact, F' is both adjacent to c_j (of tier 0) and adjacent to other cells of tier 1. There is no need for special temporary decomposition to transition between tier 0 and tier 1. F' is marked "non-refinable" until merged with F''.
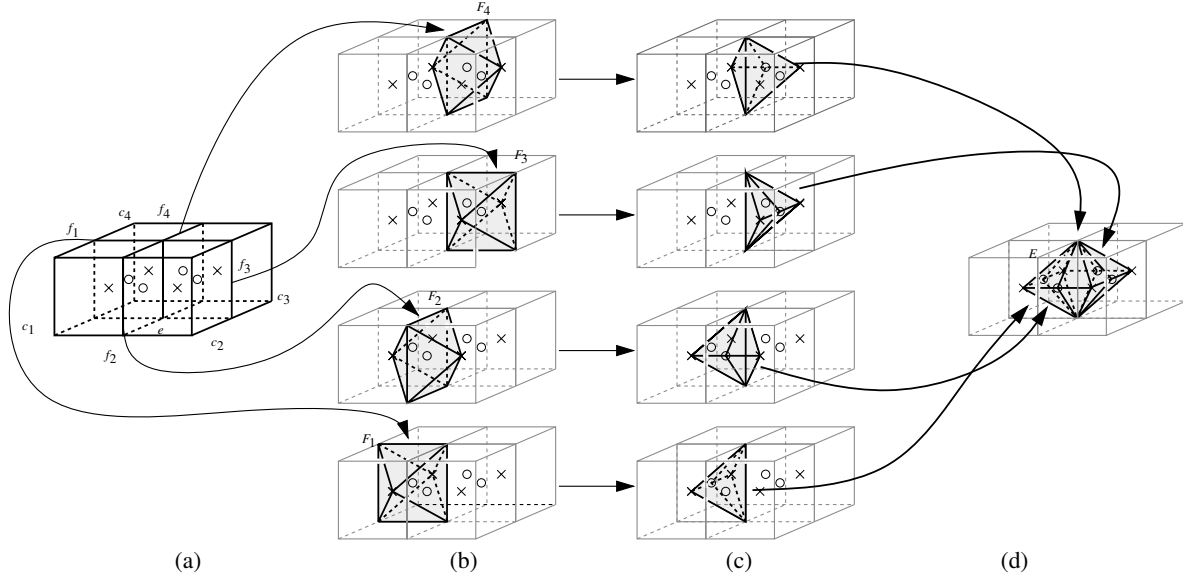
Figure 3: *Cell refinement from tier 1 to tier 2. (a) Four cells c_1, c_2, c_3 and c_4 of tier 0 share, in pairs, the facets f_1, f_2, f_3 and f_4. The edge e is shared by all facets f_1, f_2, f_3 and f_4. (b) Each facet f_i generates a cell F_i. (c) Each cell F_i is decomposed into four pyramids only two of which are selected. The selected pyramids are those containing the edge e. (c) All the pyramids containing e are merged together to form the cell E of tier 2.*
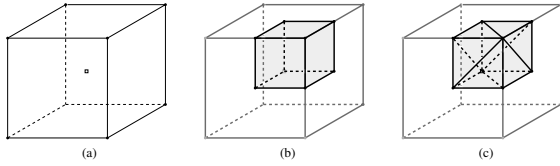

Figure 4: *Cell refinement for tier 3. (a) Cell from the mesh of tier 0. (b) The cell of tier 3 (in gray) is shown overlapped with its ancestor cell of tier 0. (c) The same refined cell with the face diagonals that are introduced during the refinement procedure.*

In the refinement from tier 1 to tier 2 we deal with sharp features of dimension one. In particular, if the merge edge e is part of a sharp feature then the pyramids around e_j are not merged but are marked as "sharp". Moreover if some of the tier-1 cells incident to e_j where not generated because of an adaptive refinement, then the pyramids are not merged but are marked as non-refinable. They are used for connecting cells of different resolution.

3.3. Characterization of the Refinement Mesh

A simple characterization can be provided for the cells generated by the *SGS* refinement.

Definition 1 A *diamond* is a cell that can be combinatorially partitioned into a set of simplices all sharing an edge, called *axis* of the diamond.

We show that all the cells generated by the *SGS* procedure are diamonds. To satisfy this property we only need each facet of the base mesh to be a polygon (a simple loop of edges).

Proposition 1 Consider a complex C where all the 2D cells are simple loops. The *SGS* procedure using C as a base mesh generates only diamond cells.

Proof. Since all the facets of the cells in C are simple loops, all the cells generated at the first tier of the subdivision procedure are either pairs of pyramids or single pyramids (for sharp features). In the first case the axis of the diamond is the the edge connecting the apices of the two pyramids. In the second case the axis connects the apex of the pyramid with the center of its base.

At the second tier the cells are diamonds by construction since they are just a set of tetrahedra merged along a common edge: the axis of the diamond.

At the third tier each cell is the set of tetrahedra sharing the edge connecting the center of a coarse cell with one of its vertices. This edge is the axis of the diamond. ◇

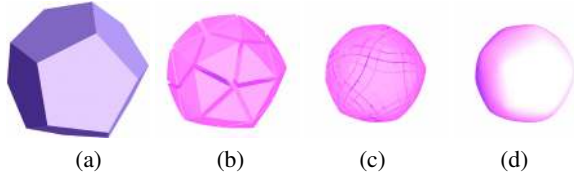Figure 5 shows the scheme applied to a base mesh consisting of a single dodecahedron.

Figure 5: *Successive subdivision stages of a dodecahedron. (a-c) Mesh elements, (d) boundary.*

4. SUBDIVISION IN ANY DIMENSION

Consider a d-dimensional mesh M with m cells $c_0, \ldots, c_{m-1}$. The complex is subdivided in $d+1$ tiers. As in the 3D case, tier d of level l is coincident with tier 0 of level $l+1$. We denote M_p^l the mesh of level l and tier p. In the refinement from tier p to tier $p+1$, one adds a vertex at the center of a $(d-p)$-dimensional face of M_0^l.

From tier p to tier $p+1$ (with $p < d-1$). Each d-cell c in M_p^l is refined by inserting a vertex q at its center. The cell c is partitioned into pyramids by connecting each of its facets to q. All the pyramids containing a common $(d-p-1)$-dimensional face of M_0^l are merged together. If one $(d-p)$-dimensional face f of M_0^l belongs to the boundary of c (there can be at most one), then q is placed at the center of f and the pyramids partitioning c are built by connecting q to all the facets of c that do not contain f.

From tier $d-1$ to tier d. Each cell c of M_{d-1}^l is split by inserting a vertex q at the center of the edge e of M_0^l contained in c. For sharp/boundary cells, e is on the boundary of c, otherwise q is also the center of c itself. The cell c is partitioned into pyramids by connecting q to the facets that do not contain e. The new cells are built by merging all the pyramids containing the same vertex of M_0^l and center of a cell of M_0^l.

Both refinement rules handle adaptivity and sharp features simply by not performing the merge step. Similarly to the 3D case we provide a simple characterization of the cells produced by the subdivision scheme.

Proposition 2 All the non-boundary/non-sharp cells generated by the subdivision scheme after the first $d-1$ tiers are d-diamonds.

Proof. The proof of follows from the equivalence of the *SGS* scheme with the d-dimensional edge bisection of Maubach [16].

First partition each cell into tetrahedra by joining each edge of the cell with the center of each incident face of higher dimension (in decreasing dimension order). If one edge e has vertices (v_0, v_1) then one builds simplices $(v_0, v_1, v_2, \ldots, v_d)$. Here v_2 is the center of a d-cell, v_3 is the center of a $(d-1)$-cell,..., and v_d is the center of a 2-cells, all incident to (v_0, v_1). Call the first edge (v_0, v_1) of the simplex the *oldest edge*. By merging all the simplices sharing

the same oldest edge one obtains the d-diamond that is created after the first $d-1$ refinements of the *SGS*.

From this stage on, the *SGS* is combinatorially equivalent to the oldest edge bisection of the simplicial complex just built. In particular each refinement is obtained by replacing the simplex $(v_0, v_1, v_2, \ldots, v_d)$ with $(v_0, v_2, \ldots, v_d, v^*)$ and $(v_1, v_2, \ldots, v_d, v^*)$ where v^* is the midpoint of (v_0, v_1). Merging all the cells with the same oldest edge one obtains the same cells as the *SGS* refinement.   *diamond*

5. DIRECT AND INDIRECT SMOOTHING

The combinatorial rules defined in the previous section need to be coupled with some averaging rules to determine the actual position of the vertices of the mesh. We propose the following simple stationary rule.

- At tier p a new vertex v inserted at the center of a cell c is the weighted average of the vertices of c, with weights dependent on their tiers:

$$v = \sum_{i=0}^{p} \sum_{j=1}^{n_p} \alpha_i w_{i,j}.$$

Here n_p is the number of vertices of c that has been created at tier p in the same level of subdivision. To reproduce the subdivision scheme of [1] it is sufficient to set $\alpha_0 = 1/n_0$ and all other $\alpha_i = 0$. In our examples we set instead all $\alpha_i = 1/\sum_{i=0}^{p} n_i$.
- Each old vertex v is repositioned by linearly combining its old position v_{old} with the average w of the vertices that are edge-connected to it.

$$v = v_{old} * \beta + (1 - \beta)w$$

To reproduce the subdivision scheme of [1] one needs to set $\beta = 0$. In most of our examples we use $\beta = 1/2$.

In the 2D case this rule is equivalent to those reported in fig. 4 of [28] since in the vertex masks the edge-connected vertices with weight 0 are exactly those that are removed by the merge stage of the recursive subdivision defined here. For lower dimensional features (boundary cases and sharp features) the weights of the terms outside such features are set to 0.

Figure 6 shows the mesh elements generated by repeatedly applying the *SGS* scheme starting from a base mesh composed of eight cubes. This subdivision is performed with an averaging coefficient $\beta = 0.5$. The parameter β can be used to alter the behavior of the smoothing mask. Figure 7 shows the successive refinements of the same mesh using two different values of $\beta = 0.89$ (left columns) and $\beta = 0.5$ (right column).

Velho and Zorin proved [28] that in the 2D case $\beta = 0.5$ yields a C^1 continuous surface with C^4 continuity at the regular points. The proof is based on the Zwart-Powell element
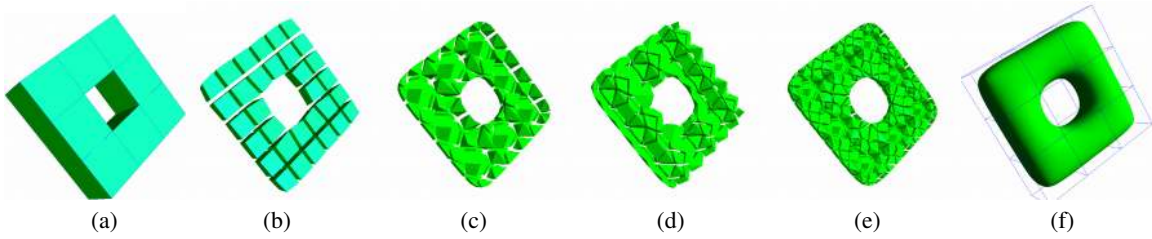
Figure 6: *Interior elements generated by the SGS scheme. (a) Base mesh formed by eight cubes. (b-e) Internal structure of the mesh in successive tiers of the SGS scheme. The pictures show only the mesh elements whose centroid has negative z coordinate. (f) Subdivided boundary mesh.*

box spline basis. Here we generalize the scheme to dimensions higher than two. This allows us to smooth the functions defined on the mesh. A more sophisticated analysis is necessary to determine the smoothness of the scheme, especially at the extraordinary points. Note that in 2D the extraordinary points can be characterized simply by their degree. In 3D, this is not sufficient since one has to analyze the local connectivity structure at the vertex (vertices with same degree can have different behavior.) Figure 8 shows the combined refinement of a mesh (in green) and isosurface (in orange) of a scalar field defined by linearly interpolating function values at the vertices of the mesh. The linear interpolation is based on a decomposition of the diamonds into tetrahedra.

6. RATE OF REFINEMENT

Consider a d-dimensional rectilinear grid, where the *SGS* scheme is equivalent to a simplicial longest edge bisection [16]. In particular, consider a rectilinear grid whose cubes are decomposed into $d!$ simplices by splitting each face along one longest diagonal. In this case, d successive steps of refinement insert a vertex at the center of each face. This is equivalent to splitting the longest edge of the simplicial decomposition d consecutive times. In particular all and only the edges of the initial simplicial decomposition are bisected. Using the terminology from Kobbelt [10], the scheme is a $\sqrt[d]{2}$ subdivision scheme. This would mean that on average the number of points in a d-grid is doubled at each refinement. To be more precise we consider an axis-aligned infinite rectilinear grid (for simplicity we do not consider boundary effects).

For $1 \leq h \leq d$, one can associate an h-dimensional face with its vertex closest to the origin. At the refinement from tier p to tier $p+1$, one vertex is added at the center of each of the $\binom{d}{p}$ faces of dimension $d-p$. Therefore, the rate of refinement is:

$$\frac{1 + \sum_{h=0}^{p} \binom{d}{p}}{1 + \sum_{h=0}^{p-1} \binom{d}{p}}$$

where we assume by convention that $\sum_{h=0}^{-1} \binom{d}{p} = 0$. For example, Figure 9 shows the 3D case. For each vertex in the

original grid a new vertex is inserted at the center of a cube, doubling the total number of vertices. At the second tier three vertices are inserted at the face centers, increasing the number of vertices by a factor of $\frac{5}{2} = 2.5$. At the third tier three vertices are used to bisect three edges, increasing the number of vertices by a factor of $\frac{8}{5} = 1.6$.

The analysis of the grid case is simplified by the fact that the total number of cells is linear in the number of vertices. This is not true in general. In particular it is known that the total number N of cells in a simplicial complex (worst case scenario) with n vertices can be as large as $N = O(n^{\lfloor d/2 \rfloor})$. The total number of vertices inserted in d tiers is $O(N)$. If after one refinement the number of vertices increases from n to βn then we obtain $\beta^d n = n^{\lfloor d/2 \rfloor}$ which implies that $\beta = O(\sqrt[d]{n^{\lfloor d/2 \rfloor - 1}})$. For $d = 2$ we have $\beta = O(1)$ confirming that in the worst case there is a constant rate of vertex increase. Even in 3D we have $\beta = O(1)$ guaranteeing a constant rate of refinement in the worst case. Interestingly for regular grids β is a small constant 2 independently of d. Note that for $d > 3$ β may not be a constant in the worst case. For example for $d = 4$ we have $\beta = O(\sqrt[4]{n})$.

7. ADAPTIVE REFINEMENT AND LOWER DIMENSIONAL FEATURES

The flexibility of the *SGS* derives from its independent treatment of cells with different dimensions. This allows one to solve two problems: (i) easy the construction of adaptive meshes, and (ii) provide explicit representation of lower-dimensional features embedded in the mesh.

The mesh adaptation between regions at different levels of resolution does not require the introduction of special temporary partitions such those required for adaptive Catmull-Clark subdivision. During the refinement procedure one simply does not perform some of the merging steps. This automatically creates the necessary adaptation. Figure 10 shows an example of mesh adaptation. The element density in the mesh increases along the positive direction of the *z*-axis.

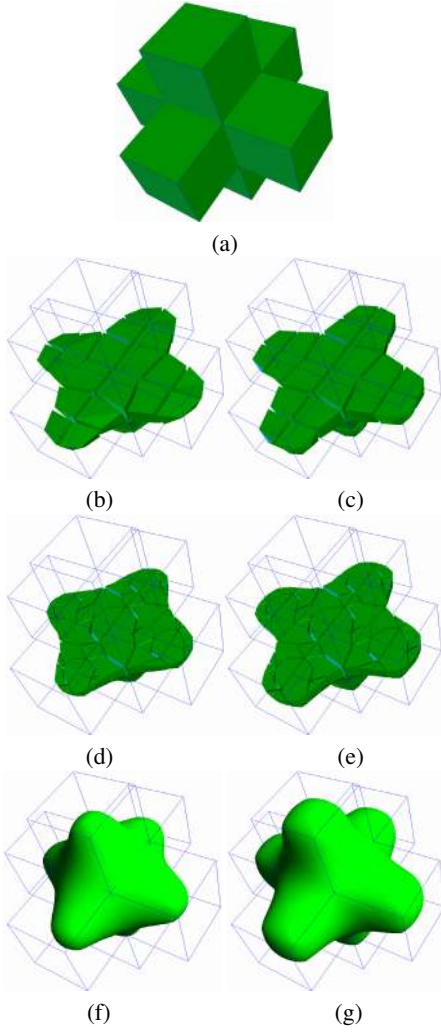A similar argument holds for lower dimensional features embedded in the mesh that the scheme automatically handles

(a)

(b)   (c)

(d)   (e)

(f)   (g)

Figure 7: *Subdivision of the same shape with different smoothing coefficient. (a) Base mesh. (b,d,f) Intermediate steps and smooth boundary for a subdivision with smoothing coefficient* β = 0.89. *(c,e,g) Same subdivision steps with* β = 0.5

in separated stages. Figure 11 shows the application of the *SGS* to a base meshes with user-defined sharp features.

For the same reason, the scheme presents no problem in handling non-manifold features as shown in Figures 12 and 13.

8. CONCLUSIONS

In this paper we have introduced the *SGS* scheme, that allows gradual refinement of a very general class of volumetric meshes. The *SGS* scheme addresses the issue of lowering the vertex proliferation rate, which is a critical problem arising
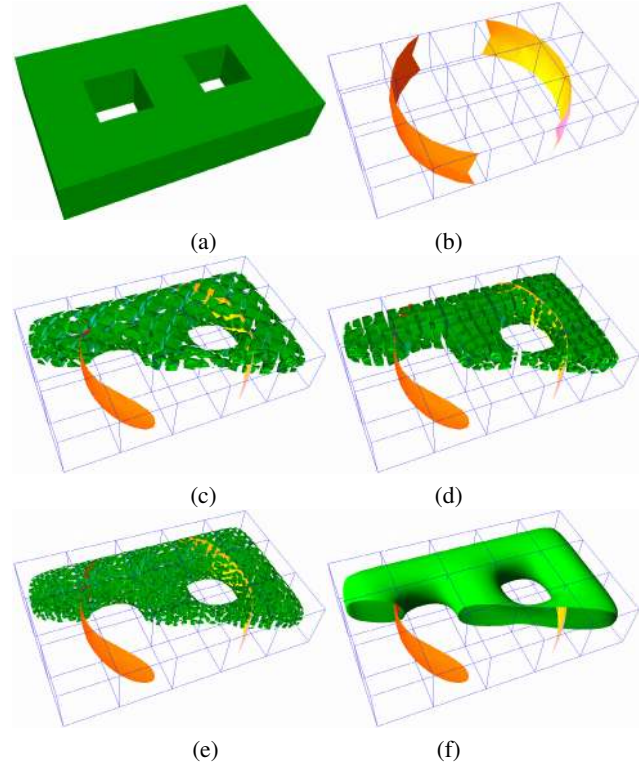

(a)   (b)

(c)   (d)

(e)   (f)

Figure 8: *Combined refinement of mesh domain and embedded isosurface. (a) Base mesh. (b) Base isosurface. (c-e) Intermediate subdivision steps. (f) Refined boundary surface and isosurface.*
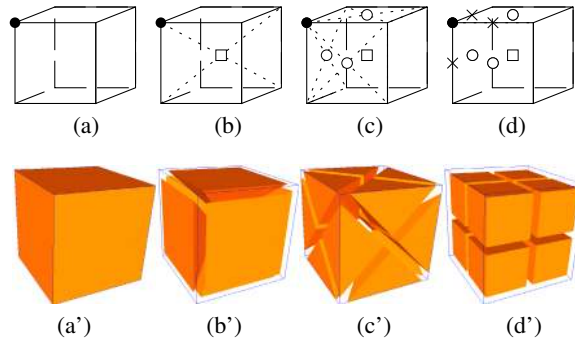

(a)   (b)   (c)   (d)

(a')   (b')   (c')   (d')

Figure 9: *(a-d) Rate of vertex increase for each refinement in a 3D rectilinear grid. For each initial vertex (marked with a black disk) one new vertex is added in the first refinement (square), three new vertices in the second refinement (circles) and three are added in the third refinement (crosses). (a'-d') Corresponding refinement of the cells in the cube.*

in the practical use of subdivision methods for volumetric meshes both for scientific visualization and geometric mod-
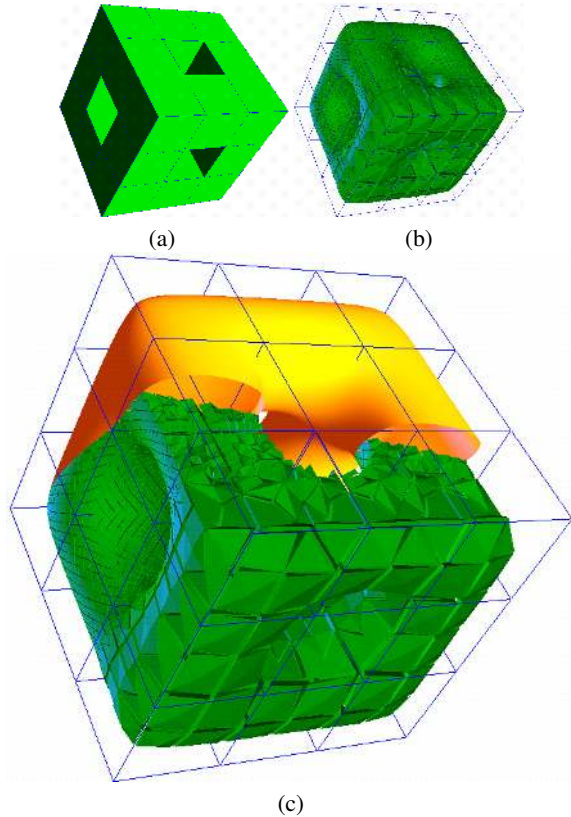
7

(a)   (b)

(c)

Figure 10: *Adaptive refinement of a 3D mesh. (a) Base mesh. (b-c) Exterior and interior refined mesh elements.*

eling. Figure 14 shows how few refinements are sufficient to produce a good meshing of a mechanical piece.

The *SGS* allows one to produce adaptive refinements of a mesh without needing special subdivision rules for cells connecting regions at different resolutions. Moreover, the scheme naturally embodies rules for handling sharp vertices, edges, and faces.

We have experimented with a very simple averaging rule that has been proven visually smooth for the 2D case and extends naturally to the volumetric and higher dimensional cases. In the 3D case we can reproduce, as a special case, the smooth subdivision of [1]. A complete analysis for the general volumetric case remains to be done as future work since it requires analysis techniques more sophisticated than those currently available.

References

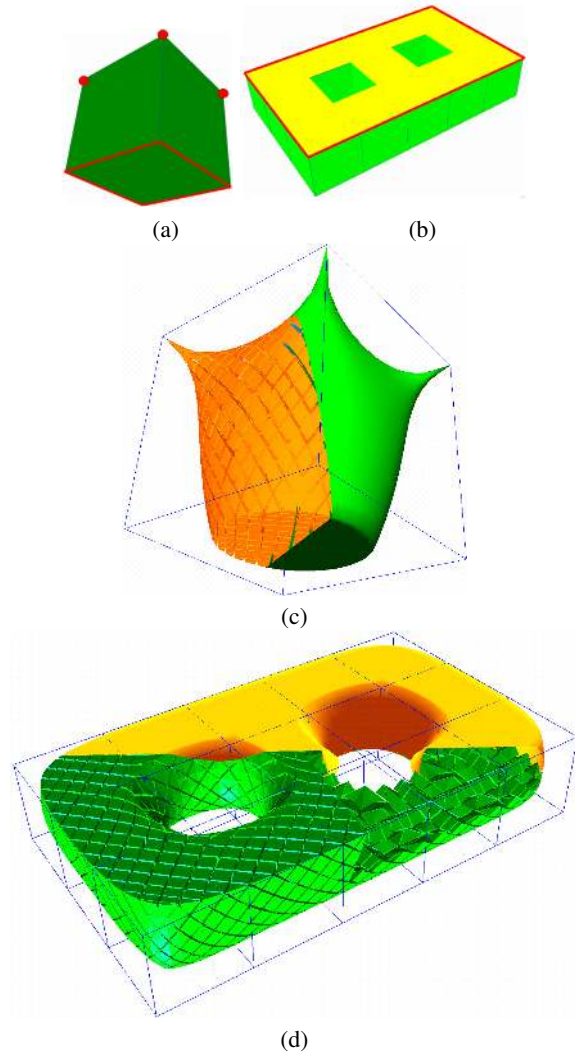1. C. L. Bajaj, J. Warren, and G. Xu. A smooth subdivision scheme for hexaedral meshes. Technical Report TICAM-REPORT-01-05, University of Texas at Austin, April 2001. Submitted for publication to "The Visual Computer".

2. R. E. Bank, A. H. Sherman, and A. Weiser. *Scientific Computing (Applications of Mathematics and Computing to the Physical Science)*, chapter Refinement Algorithms and data structures for regular local mesh refinement, pages 3–17. North Holland, 1983.
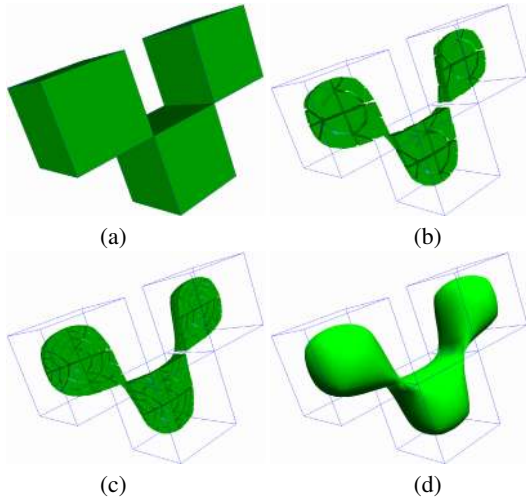
(a)   (b)

(c)

(d)

Figure 11: *Recursive subdivision with user defined sharp features. (a) Cube base mesh with the four top vertices (in yellow) marked as 0-dimensional features. The square at the bottom (in red) is marked as 1-dimensional feature. (b) Eight-shaped base mesh with a one-dimensional sharp feature (in red) and a two-dimensional feature (in yellow). (c) Refinement of the cube. (d) Refinement of the second mesh. Note the difference to the subdivision volume in Figure 8 using the same base mesh but without sharp features.*

Figure 12: *Refinement of a mesh with two non-manifold edges. (a) Base mesh. (b-c) Interior mesh of two intermediate refinement steps. (d) Smooth boundary.*
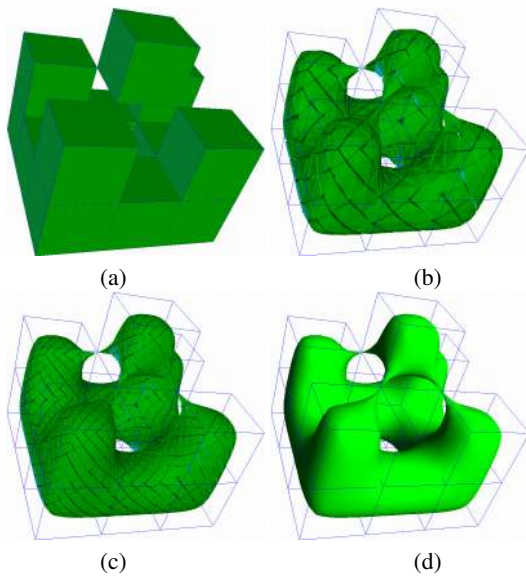

Figure 13: *Refinement of a mesh with four non-manifold points. (a) Base mesh. (b-c) Mesh elements of two intermediate refinement steps. (d) Smooth boundary.*

Figure 14: *SGS refinement of a mechanical piece. (a) Coarse mesh. (b) Mesh after six refinements (two levels). (c-h) All the six refinements for half of the mesh (cells with $z > 0$ are not shown).*

3. M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings Visualization 2000*, pages 389–396. IEEE Computer Society Technical Committe on Computer Graphics, 2000.
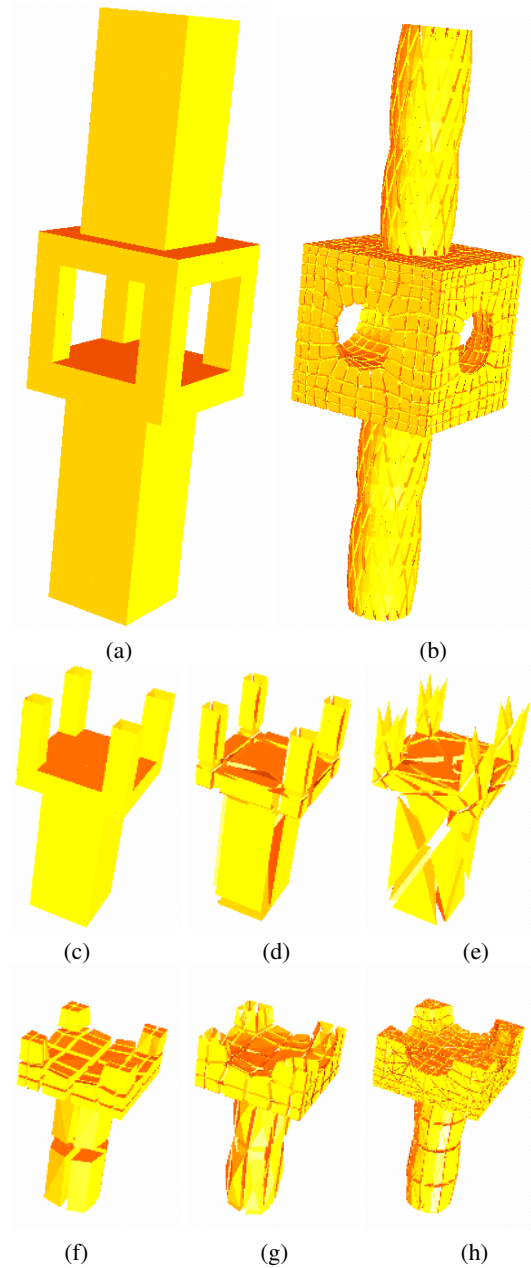
4. H. Biermann, A. Levin, and D. Zorin. Piecewise smooth division surfaces with normal control. In *Proceeding of SIGGRAPH 2000*, pages 113–120. ACM, August 2000.

5. E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological

meshes. *Computer-Aided Design*, 10:350–355, September 1978.

6. M. A. Duchaineau, B. Gregorski, and K. I. Joy. Smooth centroid bintree subdivision surfaces with local wavelets. Technical Report Technical Report, CSE-2001-1, Computer Science Department, University of California, Davis, January 2001.

7. N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. on Graphics*, 9(2):160–169, 1990.

8. M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *Computer Graphics*, 27(Annual Conference Series):35–44, 1993.

9. L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, 15(3):C409–C420, C485, September 1996.

10. L. Kobbelt. $\sqrt{3}$-subdivision. In *Proceeding of SIGGRAPH 2000*, pages 103–112. ACM, August 2000.

11. L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98 Proc.*, pages 105–114, 1998.

12. L. Linsen, J. T. Gray, V. Pascucci, M. A. Duchaineau, B. Hamann, and K. I. Joy. Hierarchical large-scale volume representation with "3rd-root-of-2" subdivision and trivariate b-spline wavelets. Technical Report CSE-2002-7, Department of Computer Science, University of California, Davis, 2002.

13. C. Loop. Smooth spline surfaces over irregular meshes. In *SIGGRAPH '94 Proc.*, pages 303–310, 1994.

14. M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.

15. R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *Computer Graphics (SIGGRAPH'96)*, pages 181–188, 1996.

16. J. M. Maubach. The amount of similarity classes created by local n-simplicial bisection refinement, 1996. Preprint.

17. S. Muraki. Multiscale volume representation by a doG wavelet. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):109–116, June 1995.

18. A. Plaza and G. F. Carey. About local refinement of tetrahedral grids based on local bisection. In *5th International Meshing Roundtable*, pages 123–136, 1996.

19. A. Raviv and G. Elber. Three dimensional freeform sculpting via zero sets of scalar trivariate functions. In *Proceedings of the fifth symposium on Solid modeling and applications*, pages 246 – 257, 1999.

20. M.-C. Rivara and C. Levin. A 3-d refinement algorithm suitable for adaptive and multi-grid techniques. *Comm. in Appl. Numer. Meth.*, 8:281–290, 1992.

21. R. Sánchez and M. Carvajal. Wavelet based adaptive interpolation for volume rendering. In *IEEE Symposium on Volume Visualization*, pages 127–134, 1998.

22. P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *SIGGRAPH 95 Proceedings*, pages 161–172. ACM, 1995.

23. O. G. Staadt, M. Gross, and R. Weber. Multiresolution compression and reconstruction. In *Proceedings of IEEE Visualization 1997*. IEEE, 1997.

24. E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgann Kaufmann, San Francisco, CA, 1996.

25. L. Velho. Using semi-regular 4-8 meshes for subdivision surfaces. *Journal of Graphics Tools*, 2001.

26. L. Velho and J. Gomes. Quasi 4-8 subdivision surfaces. In *Proceedings of SIBGRAPI'99 - XII Brazilian Symposium on Computer Graphics and Image Processing*, Campinas, SP, Brazil, October 1999. SBC - Sociedade Brasileira de Computacao, IEEE Press.

27. L. Velho and J. Gomes. Variable resolution 4-k meshes: Concepts and applications. *Computer Graphics Forum*, 2000.

28. L. Velho and D. Zorin. 4-8 subdivision. *Computer Aided Geometric Design*, 18(5):397–427, 2001.

29. D. Zorin, P. Schroder, and W. Sweldens. Interactive multiresolution mesh editing. *Computer Graphics*, 31(3A):259–268, August 1997.

30. D. Zorin, P. Schroeder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *SIGGRAPH 96 Proc.*, pages 189–192, 1996.